

Claims

We claim:

- 1 1. A method for playing frames of a video adaptively, comprising:
 - 2 measuring a spatial frequency of pixel within frames of the video;
 - 3 measuring a temporal velocity of corresponding pixels between
 - 4 frames of the video;
 - 5 multiplying the spatial frequency by the temporal velocity to obtain a
 - 6 measure of visual complexity of the frames of the video;
 - 7 playing the frames of the video at a speed that corresponds to the
 - 8 visual complexity.
- 1 2. The method of claim 1 wherein the video is compressed.
- 1 3. The method of claim 2 wherein the spatial frequency is measured from
 - 2 discrete cosine transform coefficients of the pixels in the frames, and the
 - 3 temporal velocity is measured from motion vectors of corresponding pixels
 - 4 between the frames.

- 1 4. The method of claim 3 wherein basis functions of the discrete cosine
2 transformation are in a form

$$\begin{aligned} & \cos\left(\frac{\pi k_x (2x+1)}{2N}\right) \cdot \cos\left(\frac{\pi k_y (2y+1)}{2N}\right) \\ &= \cos\left(2\pi \frac{k_x}{2N} x + 2\pi \frac{k}{4N}\right) \cdot \cos\left(2\pi \frac{k_y}{2N} y + 2\pi \frac{k}{4N}\right), \end{aligned}$$

- 3
4 where k_x is a frequency f_x in an x direction and k_y is a frequency f_y in a y
5 direction in the frame represented as

$$\cos\left(2\pi \frac{f_x}{N} x + 2\pi \frac{f_y}{N} y\right),$$

- 6
7 where N is 8 for DCT macro-blocks.

- 1 5. The method of claim 5 wherein each basis function is a superimposition
2 of two 2D sinusoids, one with a spatial frequency $\vec{f}_1 = \left(\frac{k_x}{2}, \frac{k_y}{2}\right)$ and another

- 3 with a spatial frequency $\vec{f}_2 = \left(\frac{k_x}{2}, -\frac{k_y}{2}\right)$.

- 1 6. The method of claim 5 wherein a particular motion vector is $\vec{v} = (v_x, v_y)$.

- 1 7. The method of claim 6 wherein the visual complexity resulting from the
2 discrete cosine coefficient and the motion vectors are

$$\omega_1 = \vec{f}_1 \cdot \vec{v}_1 = \frac{k_x}{2} v_x + \frac{k_y}{2} v_y, \text{ and}$$

$$\omega_2 = \vec{f}_2 \cdot \vec{v}_2 = \frac{k_x}{2} v_x - \frac{k_y}{2} v_y.$$

1 8. The method of claim 3 further comprising:
2 discarding motion vectors with a low texture;
3 median filtering the motion vectors; and
4 fitting a global motion model to the motion vectors to reduce spurious
5 motion vectors.

1 9. The method of claim 3 wherein the compressed video includes I-frames
2 and P-frames, and further comprising:
3 determined discrete cosine transformation coefficients of the P-frames
4 by applying motion compensation; and
5 determining motion vectors for the I-frames by interpolating the
6 motion vectors of the P-frames.

1 10. The method of claim 1 further comprising:
2 averaging the visual complexity over a set of frames to determine a
3 complexity of a video segment.

1 11. The method of claim 1 further comprising:
2 applying motion blur while plying the video to reduce aliasing.

1 12. The method of claim 1 wherein a speed of playing is inversely
2 proportional to the visual complexity.

1 13. The method of claim 1 further comprising:
2 applying coring to spatial filter the video while playing.

- 1 14. The method of claim 1 wherein the video is uncompressed.
- 1 15. The method of claim 1, in which a temporal distortion of the video is
2 minimized during playback.
- 1 16. The method of claim 15, in which the minimizing uses a quantization of
2 the visual complexity.
- 1 17. The method of claim 15, in which the minimizing uses a smoothing and
2 filtering of the visual complexity.
- 1 18. The method of claim 15, in which the minimizing constructs a piece-
2 wise linear approximation of the visual complexity so that the visual
3 complexity is substantially linear.
- 1 19. The method of claim 15, in which the minimizing assigns a constant
2 visual complexity to a consistent temporal segment of the video.